



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/607,119	06/26/2003	Kenneth Alexander Vadella	G&C 30566.243-US-U1	5589
55895 7590 09/21/2010 GATES & COOPER LLP HOWARD HUGHES CENTER 6701 CENTER DRIVE WEST, SUITE 1050 LOS ANGELES, CA 90045				
EXAMINER ANYA, CHARLES E				
ART UNIT		PAPER NUMBER		
2194				
MAIL DATE		DELIVERY MODE		
09/21/2010		PAPER		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte KENNETH ALEXANDER VADELLA,
J. DANIEL SMITH, and DENNIS M. MULONAS JR.

Appeal 2009-002839
Application 10/607,119
Technology Center 2100

Before HOWARD B. BLANKENSHIP, ST. JOHN COURTENAY III, and
STEPHEN C. SIU, *Administrative Patent Judges*.

SIU, *Administrative Patent Judge*.

DECISION ON APPEAL¹

¹ The two-month time period for filing an appeal or commencing a civil action, as recited in 37 C.F.R. § 1.304, or for filing a request for rehearing, as recited in 37 C.F.R. § 41.52, begins to run from the “MAIL DATE” (paper delivery mode) or the “NOTIFICATION DATE” (electronic delivery mode) shown on the PTOL-90A cover letter attached to this decision.

STATEMENT OF THE CASE

This is an appeal under 35 U.S.C. § 134(a) from the Examiner's final rejection of claims 1-24. We have jurisdiction under 35 U.S.C. § 6(b). We affirm.

Invention

The invention relates to communications between disconnected applications (Spec. 2, ¶ [0003]).

Independent claim 1 is illustrative:

1. A computer-implemented method for enabling communication between disconnected applications, comprising:

a secondary application creating a bridge object, wherein an interface for the bridge object enables communication with the secondary application through the bridge object;

registering the interface for the bridge object in a global interface table (GIT);

retrieving a cookie from the GIT in response to the registration, wherein the cookie comprises information for utilizing the interface for the bridge object; and

storing the cookie in a location that is accessible to a disconnected application such that the cookie may be retrieved to enable use of the interface, and wherein the disconnected application is unaware of the secondary application.

(Claims Appendix, App. Br. 11).

References

The Examiner relies on the following references as evidence in support of the rejection:

Srinivasan

2004/0205734 A1

Oct. 14, 2004

Microsoft Corp., *Platform SDK: COM IGlobalInterfaceTable* (2001) (“IGlobalInterfaceTable”).

Rejection

Claims 1-24 are rejected under 35 U.S.C. § 103(a) as being unpatentable over IGlobalInterfaceTable and Srinivasan.

ISSUES

Issue 1

Appellants argue that “IGlobalInterfaceTable is silent with respect to disconnected applications and the disconnected application being unaware of the secondary application” (App. Br. 6).

Issue: Did the Examiner err in finding that IGlobalInterfaceTable and Srinivasan would have taught or suggested storage of a cookie in a location accessible to a disconnected application that is unaware of a secondary application?

Issue 2

Appellants argue that “separate electronic searches of Srinivasan for the terms ‘project’ and ‘host’ provides no results whatsoever” (App. Br. 7).

Issue: Did the Examiner err in finding that IGlobalInterfaceTable and Srinivasan would have taught or suggested a secondary application that is a project hosting environment?

Issue 3

Appellants argue that “a pointer to a cookie is not a pointer to an interface” (App. Br. 7-8).

Issue: Did the Examiner err in finding that IGlobalInterfaceTable and Srinivasan would have taught or suggested registering an interface for a

bridge object in a global interface table comprising placing a pointer to the interface for the bridge object in the global interface table?

Issue 4

Appellants argue that “[i]f IGlobalInterfaceTable provides a pointer to a cookie, then the same language cannot be used to indicate that the cookie identifies the pointer and location of the interface” (App. Br. 8).

Issue: Did the Examiner err in finding that IGlobalInterfaceTable and Srinivasan would have taught or suggested a cookie that identifies a pointer and a location of an interface?

Issue 5

Appellants argue that IGlobalInterfaceTable “fails to teach, describe, or suggest the extracting of a cookie from a location that is accessible to a disconnected application” (App. Br. 8).

Issue: Did the Examiner err in finding that IGlobalInterfaceTable and Srinivasan would have taught or suggested a disconnected application extracting a cookie from a location, accessing the cookie to enable use of a bridge object, and communicating with a secondary application using an interface for the bridge object?

Issue 6

Appellants argue that “IGlobalInterfaceTable merely teaches a pointer to a ‘cookie’ that identifies an interface and its location” (App. Br. 9).

Issue: Did the Examiner err in finding that IGlobalInterfaceTable and Srinivasan would have taught or suggested storing a cookie in an environment variable?

Issue 7

Appellants argue that in the claimed invention, “while both applications are in the same process, they are unaware of each other” (App. Br. 10).

Issue: Did the Examiner err in finding that IGlobalInterfaceTable and Srinivasan would have taught or suggested a secondary application and a disconnected application executing within a same process but in different apartments?

FINDINGS OF FACT

The following Findings of Fact (FF) are shown by a preponderance of the evidence.

1. IGlobalInterfaceTable discloses “an efficient way for a process to store an interface pointer in a memory location that can be accessed from multiple apartments within the process, such as processwide variables and agile (free-threaded marshaled) objects containing interface pointers to other objects” (p. 1, ll. 8-10).
2. IGlobalInterfaceTable discloses that
[a]fter calling the CoCreateInstance function, register the interface you want to make available processwide from the apartment in which it resides with a call to the RegisterInterfaceInGlobal method. This supplies a pointer to a ‘cookie’ (through the *pdwCookie* parameter) that identifies the interface and its location. An apartment that wants a pointer to this interface then calls the GetInterfaceFromGlobal method with this cookie, and the implementation then supplies an interface pointer to the calling apartment.
(p. 1, ll. 37-41).

3. Srinivasan discloses “[a] COM [component object model] container application, such as Excel (TM) is shown as the client to receive services deployed through Jini (TM) The COM client is looking for a calculator through the Jini (TM) brokering service” (¶ [0007]).
4. Srinivasan discloses that “COM applications such as 110 cannot directly call for services from these Java objects [The object code] is wrapped as an Active X Java service” (¶ [0008]).
5. Srinivasan discloses “a first, non Java client, requesting specified services from a Java broker” (claim 1), “wherein said non Java client is a spreadsheet client” (claim 5).
6. The Specification discloses that ActiveX controls are examples of disconnected applications (p. 7, ¶ [0022]; p. 10, ¶ [0029]; p. 16-18 ¶¶ [0050-57]; p. 26, Abstract).
7. The Specification discloses that ProjectPoint is an example of “a project hosting environment” (p. 4, ¶ [0010]).
8. The Specification discloses that “[a]n example of a particular secondary application 202/project hosting environment that hosts ActiveX controls is EROOMTM (herein eRoom) (an environment available from eRoom Technology, Inc. of Cambridge Massachusetts)” (p. 11, ¶ [0033]).
9. The Specification discloses that “an environment variable 408 is an item of data that is updated by the operating system, web server or other control program. Further, the environment variable 408 resides in memory and can be read by applications” (p. 16, ¶ [0048]).

PRINCIPLES OF LAW

Claim interpretation

“In the patentability context, claims are to be given their broadest reasonable interpretations. . . . [L]imitations are not to be read into the claims from the specification.” *In re Van Geuns*, 988 F.2d 1181, 1184 (Fed. Cir. 1993) (citations omitted). Any special meaning assigned to a term “must be sufficiently clear in the specification that any departure from common usage would be so understood by a person of experience in the field of the invention.” *Multiform Desiccants Inc. v. Medzam Ltd.*, 133 F.3d 1473, 1477 (Fed. Cir. 1998).

Obviousness

The question of obviousness is resolved on the basis of underlying factual determinations including (1) the scope and content of the prior art, (2) any differences between the claimed subject matter and the prior art, and (3) the level of skill in the art. *Graham v. John Deere Co. of Kansas City*, 383 U.S. 1, 17-18 (1966). One cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. *In re Merck & Co., Inc.*, 800 F.2d 1091, 1097 (Fed. Cir. 1986).

“The combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results,” *KSR Int’l Co. v. Teleflex, Inc.*, 550 U.S. 398, 416 (2007), especially if the combination would not be “uniquely challenging or difficult for one of ordinary skill in the art,” *Leapfrog Enters., Inc. v. Fisher-Price, Inc.*, 485 F.3d 1157, 1162 (Fed. Cir. 2007) (citing *KSR*, 550 U.S. at 418).

ANALYSIS

Issue 1

Appellants challenge the Examiner's finding that Srinivasan's teachings and suggestions relate to disconnected applications. Based on Appellants' arguments in the Appeal Brief, we will decide the appeal of claims 1-24 with respect to issue 1 on the basis of claim 1. *See* 37 C.F.R. § 41.37(c)(1)(vii).

The Specification repeatedly discloses that ActiveX controls are examples of disconnected applications (FF 6). Srinivasan discloses wrapping object code as an Active X Java service (FF 4). Therefore, Srinivasan's teachings and suggestions relate to disconnected applications (Active X Java services).

Appellants argue that "the claim limitations explicitly provide that the disconnected application is unaware of the secondary application . . . [and] the COM client [of Srinivasan] is explicitly aware of a calculator" (App. Br. 6). This argument is unpersuasive because the claim language does not require that the secondary application be unaware of the disconnected application. While Srinivasan discloses a COM client (secondary application) looking for a calculator (disconnected application) (FF 3), Appellants do not show that the calculator is aware of the COM client.

Appellants further argue that "IGlobalInterfaceTable completely fails to teach, disclose, or suggest, explicitly or implicitly, any storage of a cookie in a globally accessible location" (App Br. 6). However, IGlobalInterfaceTable teaches that calling the RegisterInterfaceInGlobal method supplies a pointer to a cookie (FF 2). This cookie is accessible to other apartments, which can call the GetInterfaceFromGlobal method with the cookie to obtain a pointer to the registered interface (*id.*). Therefore, IGlobalInterfaceTable

and Srinivasan would have taught or suggested storage of a cookie (supplying a pointer to a cookie) in a location accessible (where the cookie is accessible from other apartments) to a disconnected application (Active X Java service) that is unaware of a secondary application (COM client).

For at least these reasons, we are not persuaded of error with respect to this issue in the Examiner's 35 U.S.C. § 103(a) rejections of claims 1-24.

Issue 2

Appellants challenge the Examiner's finding that Srinivasan teaches or suggests a secondary application that is a project hosting environment. Based on Appellants' arguments in the Appeal Brief, we will decide the appeal of claims 2, 8, and 14 with respect to issue 2 on the basis of claim 2. See 37 C.F.R. § 41.37(c)(1)(vii).

Appellants submit that "Srinivasan's Jini application is a stand alone application rather than an application executing in a web browser or a project hosting environment" (App. Br. 7). However, Appellants fail to submit evidence defining what constitutes a project hosting environment. The Specification does not clearly distinguish project hosting environment applications from other applications. Instead, the Specification gives two example project hosting environments, ProjectPoint (FF 7) and eRoom (FF 8), without identifying the particular project hosting environment features shared by these applications. Therefore, we rely on the plain meaning of "project hosting environment" as software that provides access (hosting environment for) to a work product (a project).

Srinivasan discloses a COM container application that looks for a calculator through a brokering service (FF 3) and that uses an Active X Java service (FF 4). Srinivasan discloses that this COM container object can be a

spreadsheet client (FF 5). A spreadsheet client is software that provides access to a work product, where the work product is a spreadsheet. Therefore, IGlobalInterfaceTable and Srinivasan would have taught or suggested a secondary application (COM container application) that is a project (spreadsheet) hosting environment (client).

For at least these reasons, we are not persuaded of error with respect to this issue in the Examiner's 35 U.S.C. § 103(a) rejections of claims 2, 8, and 14.

Issue 3

Appellants challenge the Examiner's reliance on IGlobalInterfaceTable in teaching or suggesting a pointer to an interface. Based on Appellants' arguments in the Appeal Brief, we will decide the appeal of claims 4, 5, 10, 11, 16, 17, 23, and 24 with respect to issue 3 on the basis of claim 4. *See* 37 C.F.R. § 41.37(c)(1)(vii).

Appellants submit that the claims "specify that the registering of the interface places a pointer to the interface FOR THE BRIDGE OBJECT in the global interface table" (App. Br. 7). Calling the RegisterInterfaceInGlobal method of IGlobalInterfaceTable supplies a pointer to a cookie (FF 2). Apartments can use this cookie with the GetInterfaceFromGlobal method to obtain a pointer to the registered interface (*id.*). Through these mechanisms, IGlobalInterfaceTable provides a way for a process to store an interface pointer in a memory location that can be accessed from multiple apartments (FF 1). Therefore, IGlobalInterfaceTable and Srinivasan would have taught or suggested registering an interface for a bridge object in a global interface table (calling RegisterInterfaceInGlobal) comprising placing a pointer to the interface for the bridge object (to store an interface pointer)

in the global interface table (in a memory location accessible from multiple apartments using the `GetInterfaceFromGlobal` method).

For at least these reasons, we are not persuaded of error with respect to this issue in the Examiner's 35 U.S.C. § 103(a) rejections of claims 4, 5, 10, 11, 16, 17, 23, and 24.

Issue 4

Appellants submit that `IGlobalInterfaceTable` would not have taught or suggested both a pointer to a cookie and a cookie that identified the pointer and location of an interface. Based on Appellants' arguments in the Appeal Brief, we will decide the appeal of claims 5, 11, and 17 with respect to issue 4 on the basis of claim 5. See 37 C.F.R. § 41.37(c)(1)(vii).

We are not persuaded by Appellants' reasoning. `IGlobalInterfaceTable` teaches that calling the `RegisterInterfaceInGlobal` method supplies a pointer to a cookie that identifies an interface and its location (FF 2). `IGlobalInterfaceTable` further teaches that calling the `GetInterfaceFromGlobal` method with the cookie supplies an interface pointer (*id.*). Therefore, `IGlobalInterfaceTable` and Srinivasan would have taught or suggested a cookie (supplying a pointer to a cookie) that identifies a pointer (by means of the `GetInterfaceFromGlobal` method) and a location of an interface pointer.

For at least these reasons, we are not persuaded of error with respect to this issue in the Examiner's 35 U.S.C. § 103(a) rejections of claims 5, 11, and 17.

Issue 5

Appellants argue that `IGlobalInterfaceTable` fails to teach or suggest extracting of a cookie from a location that is accessible to a disconnected

application. Based on Appellants' arguments in the Appeal Brief, we will decide the appeal of claims 6, 12, and 18 with respect to issue 5 on the basis of claim 6. *See* 37 C.F.R. § 41.37(c)(1)(vii).

IGlobalInterfaceTable teaches supplying a pointer to a cookie and an apartment using that cookie to obtain an interface pointer (FF 2). Srinivasan teaches a COM application using an Active X Java service to indirectly call for services from Java objects (FF 4). Therefore, IGlobalInterfaceTable and Srinivasan would have taught or suggested a disconnected application (Active X Java service) extracting a cookie from a location (as an apartment using a cookie), accessing the cookie to enable use of a bridge object (to obtain an interface pointer), and communicating (providing services to) with a secondary application (a COM application) using an interface for the bridge object (using the interface pointer).

For at least these reasons, we are not persuaded of error with respect to this issue in the Examiner's 35 U.S.C. § 103(a) rejections of claims 6, 12, and 18.

Issue 6

Appellants argue that IGlobalInterfaceTables' teaching of a pointer to a cookie "does not contemplate, explicitly or implicitly, an environment variable nor the storage of the cookie in such an environment variable" (App. Br. 9). Based on Appellants' arguments in the Appeal Brief, we will decide the appeal of claims 19, 21, and 23 with respect to issue 6 on the basis of claim 19. *See* 37 C.F.R. § 41.37(c)(1)(vii).

The Specification provides a broad definition of an environment variable as an item that is updated by an operating system, web server, or other control program; that resides in memory; and that can be read by

applications (FF 9). IGlobalInterfaceTable teaches supplying a pointer to a cookie that identifies the interface and its location (FF 2). This cookie can be used to obtain an interface pointer (*id.*). Thus, IGlobalInterfaceTable teaches an item (cookie) that is updated by an operating system, web server, or other control program (set to identify an interface and its location); that resides in memory (a pointer to the cookie is supplied); and that can be read by applications (and the cookie can be used to obtain an interface pointer). Because the handling of a cookie in IGlobalInterfaceTable is consistent with the Specification's disclosure regarding environment variables, we find that IGlobalInterfaceTable and Srinivasan would have taught or suggested storing a cookie in an environment variable.

For at least these reasons, we are not persuaded of error with respect to this issue in the Examiner's 35 U.S.C. § 103(a) rejections of claims 19, 21, and 23.

Issue 7

Appellants argue that IGlobalInterfaceTable fails to teach or suggest applications in the same process that are unaware of each other (App. Br. 10). Based on Appellants' arguments in the Appeal Brief, we will decide the appeal of claims 20, 22, and 24 with respect to issue 7 on the basis of claim 20. *See* 37 C.F.R. § 41.37(c)(1)(vii).

Appellants' arguments are unpersuasive because they rely on recitations that are not found in the claims. Appellants do not show that the limitation "wherein the secondary application and disconnected application are executing within a same process but in different apartments" (claim 20) requires that the applications be unaware of each other. As discussed, independent claim 1 includes the recitation "wherein the disconnected

application is unaware of the secondary application,” but this recitation does not distinguish the prior art. Appellants do not direct us to any language that requires that the secondary application be unaware of the disconnected application.

Srinivasan teaches a COM application using an Active X Java service (FF 4). IGlobalInterfaceTable teaches multiple apartments within a process (FF 1). Therefore, IGlobalInterfaceTable and Srinivasan would have taught or suggested a secondary application (COM application) and a disconnected application (Active X Java service) executing with a same process but in different apartments (in multiple apartments within a process).

For at least these reasons, we are not persuaded of error with respect to this issue in the Examiner’s 35 U.S.C. § 103(a) rejections of claims 20, 22, and 24.

CONCLUSIONS OF LAW

Based on the findings of facts and analysis above, we find no evidence persuasive of error in the Examiner’s findings:

1. that IGlobalInterfaceTable and Srinivasan would have taught or suggested storage of a cookie in a location accessible to a disconnected application that is unaware of a secondary application (issue 1);
2. that IGlobalInterfaceTable and Srinivasan would have taught or suggested a secondary application that is a project hosting environment (issue 2);
3. that IGlobalInterfaceTable and Srinivasan would have taught or suggested registering an interface for a bridge object in a global interface table comprising placing a pointer to the interface for the bridge object in the global interface table (issue 3);

4. that IGlobalInterfaceTable and Srinivasan would have taught or suggested a cookie that identifies a pointer and a location of an interface (issue 4);

5. that IGlobalInterfaceTable and Srinivasan would have taught or suggested a disconnected application extracting a cookie from a location, accessing the cookie to enable use of a bridge object, and communicating with a secondary application using an interface for the bridge object (issue 5);

6. that IGlobalInterfaceTable and Srinivasan would have taught or suggested storing a cookie in an environment variable (issue 6); and

7. that IGlobalInterfaceTable and Srinivasan would have taught or suggested a secondary application and a disconnected application executing within a same process but in different apartments (issue 7).

DECISION

We affirm the Examiner's decisions rejecting claims 1-24 under 35 U.S.C. § 103(a).

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED

rvb

GATES & COOPER LLP
HOWARD HUGHES CENTER
6701 CENTER DRIVE WEST, SUITE 1050
LOS ANGELES CA 90045